

IN THE CLAIMS:

The following claim listing replaces all previous claim listings.
Please amend claim 5 as follows:

Claim 1. (original) A method for executing a network-based distributed application, the method comprising:

executing application instances of the distributed application in application containers;

calculating quality of service metrics for each application instance;
and

distributing application workload among the application instances using a decentralized workload management layer based on the quality of service metrics.

Claim 2. (original) The method of claim 1, further comprising associating application containers with autonomous workload management elements, the workload management elements forming the workload management layer.

Claim 3. (original) The method of claim 2, further comprising coordinating the application instances through a coordination mechanism coupled to the workload management layer.

Claim 4. (original) The method of claim 1, wherein distributing application workload among the application instances further comprises reducing workload assigned to an application container when the quality of service metrics reach an overload threshold value.

Claim 5. (currently amended) The method of claim 4, wherein reducing workload assigned to the application container further comprises:

examining an encoding of work unit groups provided by each application instance;

splitting a currently assigned work unit group into smaller work unit groups;

assigning at least one of the smaller work unit groups to other application containers; and

utilizing a coordination mechanism to update changes in workload assignments to the other application containers.

Claim 6. (original) The method of claim 1, wherein distributing application workload among the application instances further comprises increasing workload assigned to the application container when the quality of service metrics reach an under-load threshold value.

Claim 7. (original) The method of claim 6, wherein increasing workload assigned to the application container further comprises:

examining an encoding of work unit groups provided by each application instance;

combining at least two currently assigned work unit groups into a smaller work unit group;

assigning the smaller work unit group to the application container; and

utilizing a coordination mechanism to update changes in workload assignments to the other application containers.

Claim 8. (original) The method of claim 1, further comprising dividing workload assigned to a single application instance to at least two application instances if a quality of service metric reaches an overload threshold.

Claim 9. (original) The method of claim 1, further comprising:

dividing a total workload performed by the distributed application among the application instances;

assigning each of the application instances a fractional workload; and

filtering client requests at the application containers based on the fractional workload assigned to the application instances.

Claim 10. (original) The method of claim 9, further comprising migrating a client from a first application container to a second application

container if workload from the client is not assigned to the application instance executing at the first application container.

Claim 11. The method of claim 10, further comprising labeling client requests such that application containers can determine if the requests belong to the fractional workload assigned to the application instances.

Claim 12. (original) The method of claim 1, further comprising receiving the application instances from application loaders.

Claim 13. (original) A system for executing a distributed computer application, the system comprising:

one or more application containers configured to execute an application instance of the distributed application and determine quality of service metrics for the application instance; and

one or more workload management elements forming a decentralized workload management layer, each workload management element is configured to be associated to one of the application containers and to assign a workload to the application container based on the quality of service metrics received by the application container.

Claim 14. (original) The system of claim 13, wherein each workload management element is further configured to autonomously increase and decrease the assigned workload to its associated application container.

Claim 15. (original) The system of claim 14, wherein each workload management element is further configured to divide the assigned workload to two or more application containers if the assigned workload to its associated application container is to be decreased.

Claim 16. (original) The system of claim 14, wherein each workload management element is further configured to combine the assigned workload of two or more application containers if the assigned workload to its associated application container is to be increased.

Claim 17. (original) The system of claim 13, wherein each application container is further configured to pass inbound packets to executing application instances when the inbound packets belong to the its assigned workload, and to pass inbound packets to its associated workload management element when the inbound packets do not belong to its assigned workload.

Claim 18. (original) The system of claim 13, further comprising workload tags coupled to data packets of application containers, the workload tags configured to allow application containers to identify whether the inbound packets belong to its assigned workload.

Claim 19. (original) The system of claim 13, further comprising a coordination mechanism configured to workload management elements to locate each other and determine the current work assignments of each application container.

Claim 20. (original) The system of claim 13, further comprising an application loader configured to provide executable application code to application containers.

Claim 21. (original) A computer program product embodied in a tangible media comprising:

- computer readable program codes coupled to the tangible media for executing a network-based distributed application, the computer readable program codes configured to cause the program to:

- execute application instances of the distributed application in application containers;
- receive quality of service metrics for each application instance; and
- distribute application workload among the application instances using a decentralized workload management layer based on the quality of service metrics.

Claim 22. (original) The computer program product of claim 21, further

comprising program code configured to associate application containers with workload management elements, the workload management elements forming the workload management layer.

Claim 23. (original) The computer program product of claim 22, further comprising program code configured to coordinate the application instances through a coordination mechanism coupled to the workload management layer.

Claim 24. (original) The computer program product of claim 21, wherein the program code configured to cause the program to distribute application workload among the application instances further comprises program code to the program to reduce workload assigned to an application container when the quality of service metrics reach an overload threshold value.

Claim 25. (original) The computer program product of claim 23, wherein the program code configured to cause the program to reduce workload assigned to the application container further comprises program code to the program to:

- examine an encoding of work unit groups provided by each application instance;

- split a currently assigned work unit group into smaller work unit groups;

- assign at least one of the smaller work unit groups to other application containers; and

- utilize a coordination mechanism to update changes in workload assignments to the other application containers.

Claim 26. (original) The computer program product of claim 21, wherein the program code configured to cause the program to distribute application workload among the application instances further comprises program code to the program to increase workload assigned to the application container when the quality of service metrics reach an under-load threshold value.

Claim 27. (original) The computer program product of claim 26, wherein

the program code configured to cause the program to increase workload assigned to the application container further comprises program code to the program to:

- examine an encoding of work unit groups provided by each application instance;
- combine at least two currently assigned work unit groups into a smaller work unit group;
- assign the smaller work unit group to the application container; and
- utilize a coordination mechanism to update changes in workload assignments to the other application containers.

Claim 28. (original) The computer program product of claim 21, further comprising program code configured to divide workload assigned to a single application instance to at least two application instances if a quality of service metric reaches an overload threshold.

Claim 29. (original) The computer program product of claim 21, further comprising program code configured to:

- divide a total workload performed by the distributed application among the application instances;
- assign each of the application instances a fractional workload; and
- filter client requests at the application containers based on the fractional workload assigned to the application instances.

Claim 30. (original) The computer program product of claim 29, further comprising program code configured to migrate a client from a first application container to a second application container if workload from the client is not assigned to the application instance executing at the first application container.

Claim 31. (original) The computer program product of claim 30, further comprising program code configured to label client requests such that application containers can determine if the requests belong to the fractional workload assigned to the application instances.

Claim 32. (original) The computer program product of claim 21, further comprising program code configured to receive the application instances from application loaders.